response, Applicant points to the following usages of "unit" in the

Specification. Those usages clearly indicate that "unit" refers

to a module of software.

> If the request is in a standard message
> format, it would invoke Message Interaction
> and Message Processing 10 and 20 layers to
> turn the message into a usable internal
> format. The Message Response Unit 100 would
> work out how to deal with the message.

(Specification, page 10, lines 4 - 6.)

> There is a Message Processing Unit that
> would carry out the actual application
> processing of the payment transaction, such as
> updating accounts and general ledger tables.

(Specification, page 10, lines 17, 18.)

> When requests are received by an
> Application Server, it would firstly try to
> understand what the request is, and then pass
> the request to the Message Processing Unit
> for processing. After processing is complete,
> the Message Response Unit would decide what
> response needs to be generated.

(Specification, page 10, lines 22 - 25.)

There are five units within this layer 3:

> 1) Communication Unit 110 - this unit is
> configurable to specify the communication
> protocol that would be used between the two
> parties. Some of the communication protocols
> supported are TCP/IP, SNA and X.25. This unit
> supports both external communication as well
> as Inter-Process Communication (IPC).

2) Middleware Support **Unit** 115 - Some banks may decide to use a Middleware product to implement their payment systems. The middleware product has in-built capabilities to guarantee delivery of payment messages and provide high availability features. In addition, some middleware products also provide XA transaction processing compliance transaction manager. This unit could integrate into the Payment Switch seamlessly over the chosen communication protocol.

3) File Interface Support **Unit** 120 - this unit is responsible for interfacing with files, such as locating an input file, and opening, reading, and writing to a file. It is also responsible for sending and receiving files. Since some payment systems deal with transactions at the file level instead of the message level, it is necessary to include this unit 120 at this level. lowing easier modification of the system and maximum flexibility.

4) Timer **Unit** 125 - this unit is used to handle trigger mechanisms when a payment transaction is ready. The mechanism could also be an alarm clock that signals time for the process to initiate a read from some data source.

5) Database Interface **Unit** 135 - this is the unit where physical access of data is carried out. Programmers must program within this unit to indicate how to read the data.

All units would work together to provide **a communication layer of software** which is totally transparent to the calling software.

(Specification, page 12, lines 8 - 29.)

There are six units within the Message Interaction Layer 10. They represent different aspects of the message FAP. The entities exchanging messages would have to

3

understand FAP through these units.

1) Message Format **Unit** 150 - this unit defines message formats that are exchanged between entities such as Participants, Payment Switch and Payment Servers. The message format can either be an external message or an internal message. The unit 150 understands the format transformation between an external and internal message.

2) Message Response **Unit** 155 - this unit defines for each internal and external message when processed by an entity, what message response(s) is required. For instance, upon receiving a payment instruction, the Payment Switch would respond by an acknowledgment.

3) Message Synchronicity **Unit** 160 - This unit controls the message delivery and response behaviour of an entity. For the sender, whether it will wait for a response after sending a message. For a receiver, whether it will respond immediately.

4) Message Packaging **Unit** 170 - this unit is responsible for packaging external messages for transportation and interpretation. One or more external messages may be packaged into a single buffer for read/write. A message can also be a file handle to access a batch file.

5) Message Initiation **Unit** 175 - this unit defines how participants of a payment system transfer messages. The participants may work in a client/server model where the client always initiates to send messages or initiates to retrieve messages (PULL Model). The participants may work in a co-operative manner where both parties deliver messages to other participants in an asynchronous manner (PUSH Model).

6) Message Routing **Unit** 180 - this unit defines the route of a message (internal or external) once a response is to be initiated.

4

Each message that travels outside of an entity
would require the routing information to be
associated with it so that the entity
understands where to send the message. The
internal route could be a server ID and the
external route could be a participant ID.

These units within the Message
Interaction Layer 10 have a well defined
interface with Message Control Module 30.

(Specification, page 13, line 22 - page 14,
line 21.)

Part of the software, such as Message
Translation Unit, may also be ported to NT 4.0
environment, but at this stage, there is no
plan to have EPSW run on NT until version 2.0.

(Specification, page 18, lines 9 - 11.)

See also Specification, page 15, lines 3 - 26; page 16, line

7 - page 17, line 12.

Therefore, the preceding passages of the Specification clearly

indicate that the Specification uses the term "unit," as in

"Message Translation Unit," to refer to a piece of software. MPEP

§ 2173.01 states:

[A]pplicants are their own lexicographers.

They can define in the claims what they regard
as their invention essentially in whatever
terms they choose so long as the terms used
are not used in ways that are contrary to
accepted meanings in the art.

. . .

A claim may not be rejected solely because of
the type of language used to define the

5

subject matter for which patent protection is
sought.

MPEP § 2173.02 states:

CLARITY AND PRECISION

(End of first paragraph) Examiners . . .
should not reject claims or insist on their
own preferences if other modes of expression
selected by applicants satisfy the statutory
requirement.

. . .

Definiteness of claims language must be
analyzed, **not in a vacuum**, but in light of (1)
the content of the particular application
disclosure . . .

Finally, Applicant points out that a similar term, namely,
"object" is widely used. The term "object" is used in, for
example, the documentation explaining the languages C and C++.
"Object" can refer to a software module. Yates uses "object" in
this sense: column 17, line 16.

Applicant requests an explanation of why "object" is
acceptable, and yet "unit" is not. In fact, one definition of
"unit" is "a single thing, or object."

Application requests a citation of authority in support of the
objection.

Further, the objection is based on a false analysis.

    -- The claims refer to "software unit A" and

"software unit B."

-- The objection asserts that the **same term**

is being used to refer to two **different** claim

elements.

The latter statement is false. The terms are **different**. The term

"software unit A" is different from "software unit B."

The same term is not being used.

On August 5, 2004, the undersigned attorney did a search for

the phrase "software unit" on the PTO's web site, in patents issued

since 1976. Three hundred thirty-seven hits were obtained.

For example, patent 6,771,668 states:

> In a **SOFTWARE UNIT** 802, an application layer
> 816, differs in software used by the system,
> and the data transfer protocol indicating how
> to transfer data on the interface is defined
> by a protocol such as a printer protocol or an
> AVC protocol.
>
> (Tenth paragraph of Detailed Description of
> Invention, in Internet version.)

Applicant requests an explanation of why those 337 patents can

use the term "software unit," yet Applicant is prohibited from

doing so.

re: Office Action, Page 2, section 4

Applicant requests a citation of authority in support of the

suggestion of modifying the drawings. One reason is that it is

7

axiomatic that new matter cannot be added to an application. Another reason is that the undersigned attorney is aware of no requirement that the drawings must "further clarify" the claims.

"Further clarify" means that clarity already exists. Why is Applicant required to supply additional, that is, redundant, clarity ?

re: Office Action, Page 2, section 5

As to arguing patentability of new claims, Applicant states: each added dependent claim is considered patentable, because of patentability of the parent.

In addition, added claim 8 recites "c) installing the software systems into electronic payment switches." The applied references do not show the overall recitations of this claim, including this recitation.

Added claim 9 recites "installing the software systems into electronic payment switches." The applied references do not show the overall recitations of this claim, including this recitation.

Added claim 10 recites "installing the software system into an electronic payment switch." The applied references do not show the overall recitations of this claim, including this recitation.

Added claim 11 recites "installing the software system into an electronic payment switch." The applied references do not show the overall recitations of this claim, including this recitation.

8

Added claim 12 recites

> c) repeating steps of paragraph (b) to
> thereby modify a software system previously
> constructed; and
>
> d) installing the modified software system
> into an electronic payment switch.

The applied references do not show the overall recitations of this claim, including this recitation.

Added claim 13 recites:

> c) repeating steps of paragraphs (a) and (b)
> to thereby modify a software system previously
> fabricated; and
>
> d) installing the modified software system
> into an electronic payment switch.

The applied references do not show the overall recitations of this claim, including this recitation.

MPEP 2143.03 states:

> To establish _prima facie_ obviousness . . . **all
> the claim limitations** must be taught or
> suggested by the prior art.

Since the claim recitations identified above have not been shown in the applied references, the 103-rejections cannot stand.

re: Office Action, Page 2, section 6

Objection was registered to the use of sub-headers.

Sub-headers are used, as in claim 8, which states:

> 8.   Method according to claim 1, and further comprising the step of
>
> c)   installing the software systems into electronic payment switches.

The sub-header "c)" is used to identify the paragraph, since the

last paragraph in parent claim 1 is labeled "b)".

Applicant requests a citation of authority in support of the

objection.

Applicant further points to US patent 5,577,734 (Etzel et al.,

June 10, 2003, 08/550,909), wherein claim 5 states:

> 5.   Method according to claim 4, and further comprising the steps of:
>
> b)   effectively transmitting a stored encrypted key from one sub-system to another,
>
> i)   de-crypting the encrypted key into plain text,
>
> ii)   encrypting the plain text into cypher text, using a transmission key, and
>
> iii)   transmitting the cypher text on a communication channel.

See also claims 7 - 10 in Etzel.

Applicant requests an explanation of why the sub-headers are

acceptable in Etzel, but not in the present case.